ARMY RESEARCH LABORATORY

An Implementation of a Dynamic Synthetic Environment Using the Silicon Graphics Performer Graphics Library

by Mark A. Thomas

ARL-TR-1454 August 1997

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5067

ARL-TR-1454 **August 1997**

An Implementation of a Dynamic Synthetic Environment Using the Silicon Graphics Performer Graphics Library

Mark A. Thomas
Information Science and Technology Directorate, ARL

Abstract

Synthetic, virtual environments are graphical representations of physical reality that humans can interact with using a variety of computer interfaces. A synthetic, virtual environment might be a building interior, an entire city, a large land area, or even a complete fictional world. High-speed computer graphics, combined with supercomputer processing power, allow the development of high-detail, realistic environments for training and entertainment. The processing power of supercomputers allows the computation of real-world, physical phenomena such as terrain deformation, atmospheric clouds, smoke plumes, and the physical effects caused by the turbulence and irregularities in the atmosphere. Traditional graphics implementations of these phenomena require the use of low-level graphical objects, voxels, or rotating texture maps, to render these phenomena. This report presents an implementation of Fractal Ellipsoids to represent smoke plumes using the Silicon Graphics (SGI) Performer Graphics Library. In addition, the representation of deformable, changeable terrain using the ARL Variable Resolution Terrain model is explained. The applicability of these techniques provides a powerful mechanism to merge high-speed, realistic graphics rendering with the real-time data processing and computational power of supercomputers, which will provide synthetic environments with real-world characteristics.

ACKNOWLEDGMENTS

The author would like to thank Terry Purnell and Weiqun Zhou for implementing the dynamic terrain and smoke servers used in this system.

INTENTIONALLY LEFT BLANK.

TABLE OF CONTENTS

INTENTIONALLY LEFT BLANK.

LIST OF FIGURES

INTENTIONALLY LEFT BLANK.

1. INTRODUCTION

Synthetic, virtual environments are graphical representations of real or imaginary places. The technology of virtual reality has been used for years in simulators for training, mission rehearsal, and disaster reconstructions.

The increasing speed and processing power of graphics supercomputers provide realistic drawings at frame rates above 30 Hz for simple geometries. However, the processing required for realistic depiction of atmospheric and dynamic phenomena requires the use of computationally expensive algorithms. The result is a realistic rendering at the expense of frame rate.

The Silicon Graphics (SGI) Performer Library is a graphics programming interface that provides multiprocessing to separate the application processing from the actual graphics rendering (Hartman and Creek 1994).

The use of multiprocessing allows the application developer to separate the nongraphical application code from the drawing process. This provides a means to tune the simulation for varying graphical and computational environments. However, the lack of computing resources remains a problem because of the increasing demand for more realistic training environments composed of high-definition terrain; atmospheric effects that have real-world characteristics for use with sensors and night-vision devices; environmental cues; and accurate targeting and damage assessment.

The solution is to separate the graphics application from performing complex and compute-intensive processing. The graphics application then would only have to draw graphics primitives and not have to process complex algorithms for atmospheric phenomena, dynamic environment changes, or complex, dynamic moving models. Supercomputers and workstations can be used to compute and manipulate graphic primitive data such as vertex locations, normals, and texture positions.

This report presents a method to interface the graphics engine to workstations to create complex scenes with dynamic atmospheric features and terrain in real time.

2. DYNAMIC ENVIRONMENTS ARCHITECTURE

The ability to deform terrain is important to ground-based systems. Indentations which are ignored by airborne systems can be used to provide cover or give clues to the movement of vehicles for ground-based observers or to interfere with line-of-sight for ground-based observers.

To provide dynamic deformation, the U.S. Army Research Laboratory (ARL) has developed mathematical models of terrain (Purnell et al. 1995). These models provide a real-time computational mechanism to create and modify terrain. The data output of these models consists of the vertex, normal, and texture coordinates for the surface. To implement the dynamic terrain in the graphics system requires an interface which provides for the rapid update of the graphics database, yet separates the terrain model computations from the graphics engine.

The interface is provided by a shared-memory connection between the terrain model and the graphics engine. The shared-memory is used to store the vertex, normal, color, and texture data required for the terrain. The graphics engine simply loads these memory pointers into main memory and draws it.

The components of Figure 1 illustrate the division of labor to produce a high-detail synthetic environment. The workstation/supercomputer compute nodes perform all the processing for modifying environmental attributes. These attributes may include the emissive properties of a dust cloud, the shadow of cloud cover, or the deformation of terrain due to explosions. The graphics engine draws the high-detail scenery.

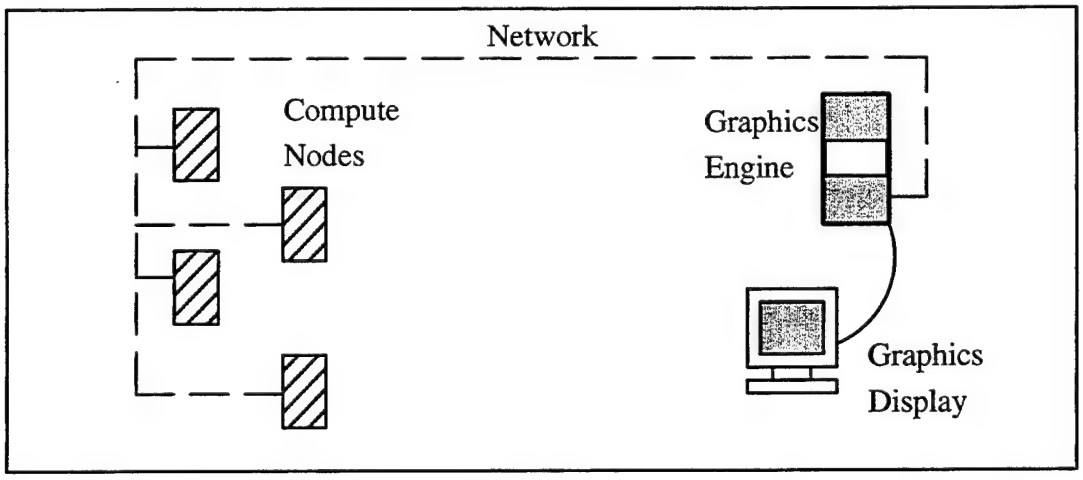

Figure 1. Dynamic environments hardware components.

The software architecture on the graphics engine provides the link between the data produced by the compute nodes and the graphics software database. Figure 2 illustrates the components of the software for dynamic environments. The terrain_server and smoke_manager are the external programs to the graphics program, STEALTH. The terrain_server receives vertex information over the network and writes it to the shared memory area. The smoke_manager performs the same task for smoke. This method works because the vertex data for the renderer are accessible both by the graphics program and the server. Therefore, either program can modify the data.

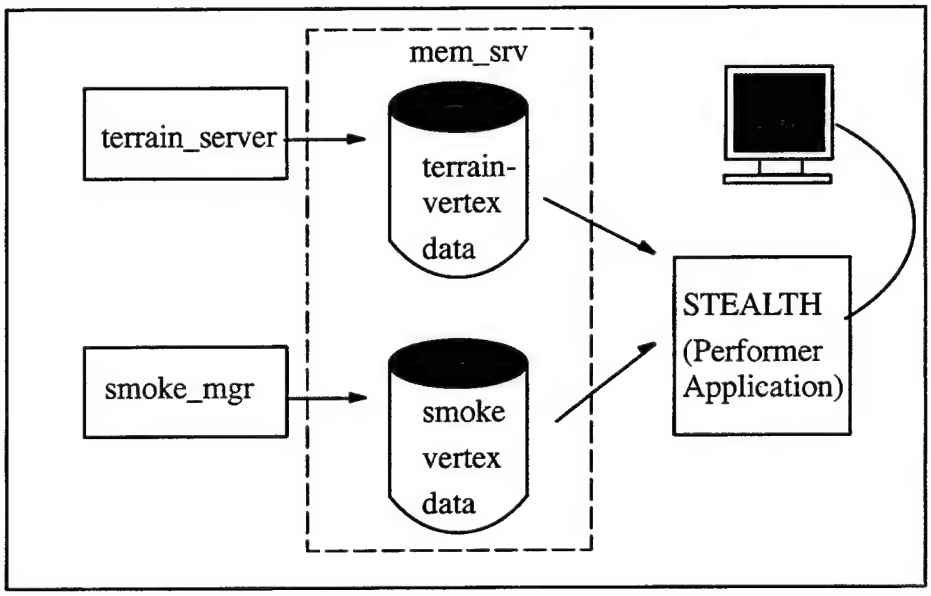

Figure 2. Graphics engine software components.

The mem_srv program creates the virtual memory for the system. This memory is a disk file that looks like main memory to applications which address it. It allows for memory locking for data access management. A third-party program is used to ensure that the virtual memory addresses of the STEALTH and the client programs reside at the same memory addresses. If not, the memory allocation will probably fail (Silicon Graphics, Inc. 1994).

The mem_srv allocates dynamic memory using the Datapool memory functions of Performer. The datapool memory size is set by command-line flags. In its present form, the mem_srv program simply allocates memory, then sleeps (see Figure 3).

```
main( argc, argv )
{
  parse_commandline();
   for( i = 0; i < NUM_DATA_POOLS; i++ ){
      datapool[i] = pfNewDPool( DatapoolSize[i], DatapoolName[i] );
   }
   sleep( 28800 );
}
```

Figure 3. Memory server program pseudocode.

3. DYNAMIC TERRAIN DATA STRUCTURE

The data structure for implementing dynamic terrain consists of a header, followed by vertex data. The header provides a handshake between the STEALTH and the terrain_server to provide synchronization, data memory requirements, and the addresses of the coordinate, normal, color, and texture memory. Synchronization is required to provide an update to the database collision detections for objects that require height-above-ground information. Terrain height updates ensure that ground-based moving objects descend into craters and not float over them. Figure 4 shows the data structure used for dynamic terrain.

The terrain data structure components are suitable for use in Performer data structures. The coordinate, texture, color, normal, gset, and index structure components are pointers to virtual memory locations; therefore, they can be cast to the correct data type for use in Performer functions. Figure 5 shows a pseudocode example of creating a database using the dynamic terrain structure.

The vertex memory is allocated in the terrain_server program. The program exploits the indexed tristrip capability of SGI Performer (Hartman and Creek 1994). Indexed tristrips are graphical primitives that create objects using collections of triangles (TRISTRIP) (Figure 6). The TRISTRIP is composed of n-2 triangles, where n is the number of vertices in the TRISTRIP.

```
#define TERRAIN_SERVER_INIT   0
#define TERRAIN_UPDATE_ALL  1
#define TERRAIN_UPDATE_SEGMENT 2
struct _gset{
   long num_prims;
   long index;        /* address of the index list for this geoset */
};
struct _t_str{
   long status;                        /* Terrain database status    */
   long utime;                         /* Last update time of this terrain */
   long ctime;                         /* Create time of this terrain    */
   long reason;
   long num_geosets;                   /* The number of geosets in this terrain */
   long vertices;                      /* Address of the vertex coordinates */
   long texcoords;                     /* Address of the texture coordinates */
   long normals;                       /* Address of the normal coordinates */
   long colors;                        /* Address of the color memory    */
   struct _gset *gset;                 /* Address of the geoset data      */
};
```

Figure 4. Dynamic terrain data structure.

The use of indexed TRISTRIPS requires an index of vertices to be created for each TRISTRIP. The index list for each TRISTRIP is unique, allowing the reuse of the coordinate, texture, and normal memory pointers.

The terrain_server, upon initialization, allocates all memory for the terrain being modeled. It then copies the database pointers into the dynamic terrain data structure and waits for the database updates from the Variable Resolution Terrain Manager (Purnell et al. 1995). The terrain_server notifies the STEALTH that data have changed by setting the flag struct_t_str.reason to TERRAIN_UPDATE_ALL. This causes the STEALTH to recompute collision detection with the terrain, so updates are reflected in dynamic model behavior, such as a tank descending into a crater. A representative crater is illustrated in Figure 7, and the initial ground is shown in Figure 8.

```
/* Get the shared memory pointers    */
alist = (pfVec3 *)tsrv–>vertices;
nlist = (pfVec3 *)tsrv–>normals;
clist = (pfVec4 *)tsrv–>colors;
tlist = (pfVec2 *)tsrv–>texcoords;

/* Create the index list, one per geoset    */
for( i = 0; i < tsrv–>num_geosets; i++ ){
     idx = CreateIndexForGeoset;

    /* Create the graphical object */
    geoset = pfNewGSet( arena );
   pfGSetPrimType(geoset, PFGS_TRISTRIP );
    pfGSetNumPrims(geoset, 1 );
    length = (long *)pfMalloc( sizeof( long ), arena );
    length[0] = 2 * numprims;
    pfGSetPrimLengths( geoset, length );
    pfGSetAttr( geoset, PFGS_COORD3, alist, idx );
   pfGSetAttr( geoset, PFGS_TEXCOORD2, tlist, idx );
    pfGSetAttr( geoset, PFGS_COLOR4, clist, idx );
    pfGSetAttr( geoset, PFGS_NORMAL3, nlist, idx );
 }
```

Figure 5. Terrain database creation in SGI Performer.

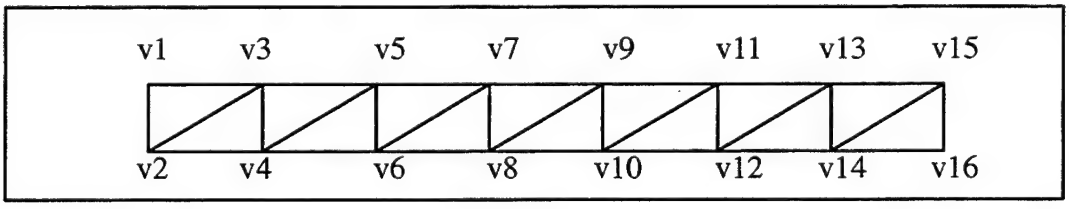

Figure 6. Triangle strip (TRISTRIP).

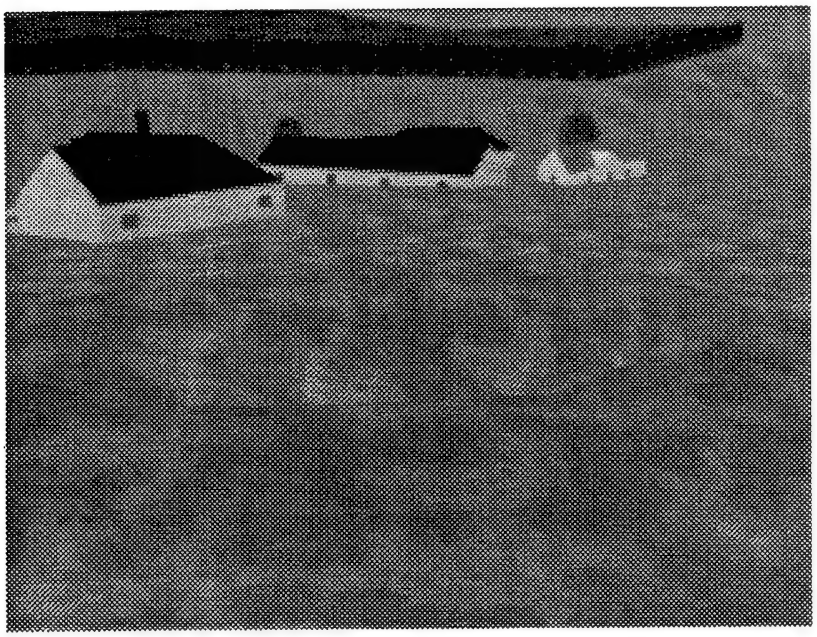

Figure 7. Ground with crater.

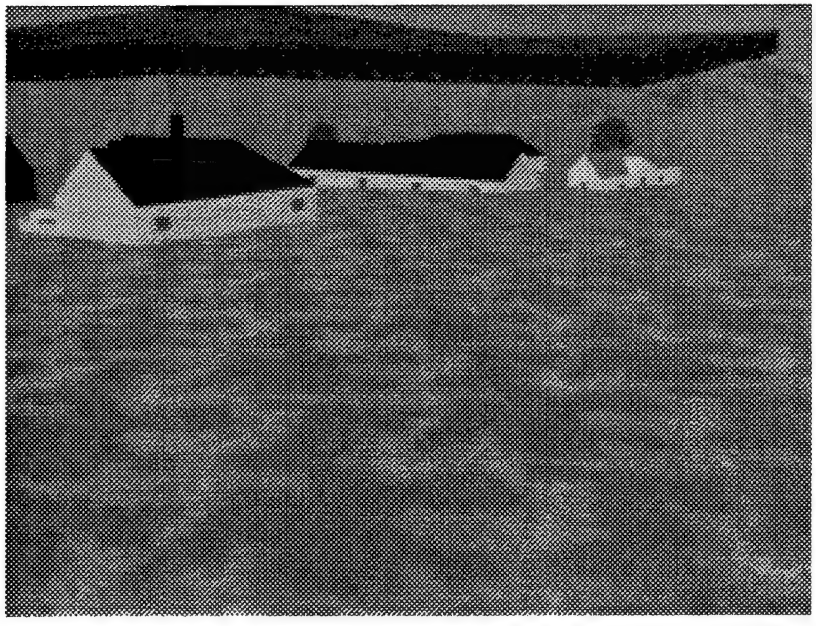

Figure 8. Initial ground.

4. REAL-TIME SMOKE PLUMES

The rendering of realistic smoke plumes using computer graphics is a problem that has been solved by various researchers using a variety of methods (Gardner 1994). A popular method of rendering smoke plumes uses volume elements, or voxels, to render the plumes. An implementation of the COMBIC smoke model, using Fractal Ellipsoids (Gardner 1994), is used in the STEALTH. The algorithm uses fractals to compute the smoke plume density at each vertex, and these intensities are treated as alpha values in the color of the vertex (Gardner 1994).

The STEALTH implements smoke plumes using smoke textures. At program initialization, smoke plume objects are loaded from disk into texture memory. When smoke is required, the plume objects are added to the scene. All smoke plume computation is performed offline by the smoke_manager program. The graphics renderer simply draws it.

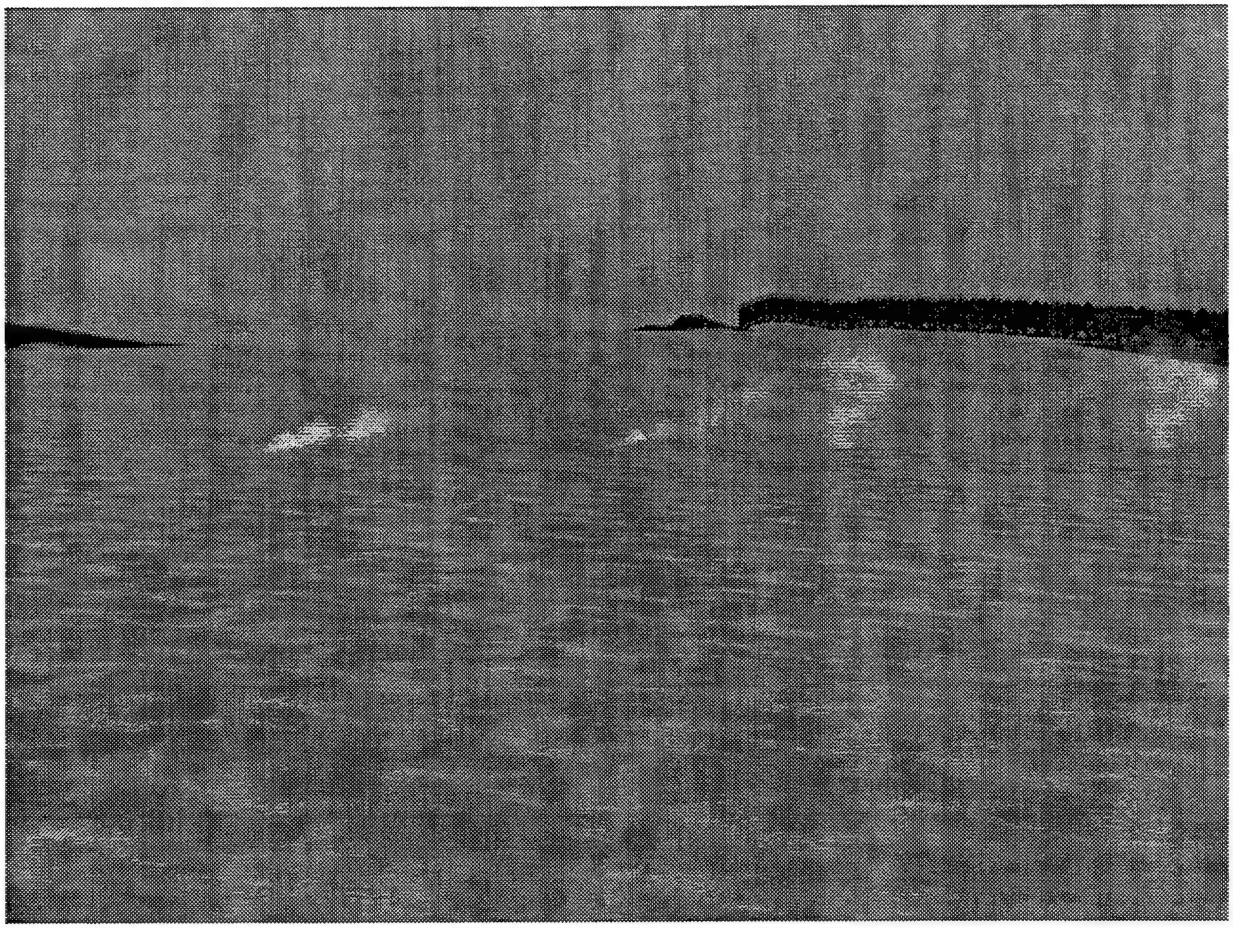

Figure 9. Multiple smoke plumes.

5. CONCLUSIONS

The shared-memory technique described in this report provides a powerful mechanism for displaying high-resolution, dynamic terrain and atmospheric effects in a real-time simulation system. The technique allows the use of the SGI Performer Library, with its virtual environment development capabilities, to be utilized without losing the ability to insert novel dynamic features.

The separation of the graphics engine from the dynamic animation computations allows the graphics engine to draw complex geometry, eliminating the computation bottleneck. Graphics load can be tuned for specific requirements.

This technique can be extended to provide animation for moving water, building berms, or other incremental animations.

The technique here animates a fixed-length tristrip. Future versions will allow for variable-length tristrips.

Currently, a single terrain node is used during a simulation. In the future, the ability to use multiple dynamic terrain models simultaneously will be implemented.

INTENTIONALLY LEFT BLANK.

6. REFERENCES

Gardner, G. Y. "Modeling Amorphous Natural Features." Course notes, SIGGRAPH, Orlando, FL, 1994.

Hartman, J., and P. Creek. IRIS Performer Programming Guide. Silicon Graphics, Inc., 1994.

Purnell, T., T. Kendall, R. Pearson, W. Zhou, and W. Qiu. "Implementation of Variable Resolution Geospecific Terrain in the ARL Individual Soldier Distributed Interactive Simulation Test-Bed." Proceedings of the 1995 Simulation MultiConference, 9–13 April, Phoenix, AZ, 1995.

Silicon Graphics, Inc. USINIT (3P) On-Line Reference Manual. Silicon Graphics, Inc., 1994.

INTENTIONALLY LEFT BLANK.

NO. OF COPIES	ORGANIZATION
2	DEFENSE TECHNICAL INFORMATION CENTER DTIC DDA 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
1	HQDA DAMO FDQ DENNIS SCHMIDT 400 ARMY PENTAGON WASHINGTON DC 20310-0460
1	CECOM SP & TRRSTRL COMMCTN DIV AMSEL RD ST MC M H SOICHER FT MONMOUTH NJ 07703-5203
1	PRIN DPTY FOR TCHNLGY HQ US ARMY MATCOM AMCDCG T M FISETTE 5001 EISENHOWER AVE ALEXANDRIA VA 22333-0001
1	PRIN DPTY FOR ACQUSTN HQS US ARMY MATCOM AMCDCG A D ADAMS 5001 EISENHOWER AVE ALEXANDRIA VA 22333-0001
1	DPTY CG FOR RDE HQS US ARMY MATCOM AMCRD BG BEAUCHAMP 5001 EISENHOWER AVE ALEXANDRIA VA 22333-0001
1	DPTY ASSIST SCY FOR R&T SARD TT T KILLION THE PENTAGON WASHINGTON DC 20310-0103
1	OSD OUSD(A&T)/ODDDR&E(R) J LUPO THE PENTAGON WASHINGTON DC 20301-7100

NO. OF COPIES	ORGANIZATION
1	INST FOR ADVNCD TCHNLGY THE UNIV OF TEXAS AT AUSTIN PO BOX 202797 AUSTIN TX 78720-2797
1	DUSD SPACE 1E765 J G MCNEFF 3900 DEFENSE PENTAGON WASHINGTON DC 20301-3900
1	USAASA MOAS AI W PARRON 9325 GUNSTON RD STE N319 FT BELVOIR VA 22060-5582
1	CECOM PM GPS COL S YOUNG FT MONMOUTH NJ 07703
1	GPS JOINT PROG OFC DIR COL J CLAY 2435 VELA WAY STE 1613 LOS ANGELES AFB CA 90245-5500
1	ELECTRONIC SYS DIV DIR CECOM RDEC J NIEMELA FT MONMOUTH NJ 07703
3	DARPA L STOTTS J PENNELLA B KASPAR 3701 N FAIRFAX DR ARLINGTON VA 22203-1714
1	SPCL ASST TO WING CMNDR 50SW/CCX CAPT P H BERNSTEIN 300 O'MALLEY AVE STE 20 FALCON AFB CO 80912-3020
1	USAF SMC/CED DMA/JPO M ISON 2435 VELA WAY STE 1613 LOS ANGELES AFB CA 90245-5500

NO. OF
COPIES ORGANIZATION

1 US MILITARY ACADEMY
MATH SCI CTR OF EXCELLENCE
DEPT OF MATHEMATICAL SCI
MDN A MAJ DON ENGEN
THAYER HALL
WEST POINT NY 10996-1786

1 DIRECTOR
US ARMY RESEARCH LAB
AMSRL CS AL TP
2800 POWDER MILL RD
ADELPHI MD 20783-1145

1 DIRECTOR
US ARMY RESEARCH LAB
AMSRL CS AL TA
2800 POWDER MILL RD
ADELPHI MD 20783-1145

3 DIRECTOR
US ARMY RESEARCH LAB
AMSRL CI LL
2800 POWDER MILL RD
ADELPHI MD 20783-1145

ABERDEEN PROVING GROUND

2 DIR USARL
AMSRL CI LP (305)

NO. OF COPIES	ORGANIZATION
1	OFC OF THE ASST SECRETARY RSRCH DEV AND ACQUISTITION G T SINGLEY III 103 ARMY PENTAGON WASHINGTON DC 20310-0103
1	US ARMY STRICOM PM DIS AMCPM DIS CENTRAL FLORIDA RSRCH PARK 12350 RESEARCH PKWY ORLANDO FL 32826-3276
1	US ARMY STRICOM AMSTI ET TRACI JONES CENTRAL FLORIDA RSRCH PARK 12350 RESEARCH PKWY ORLANDO FL 32826-3276
1	COMMANDANT ATSH CDA DISMOUNTED BATTLESPACE BATTLE LAB USIS FORT BENNING GA 31905-5400
1	UNIV OF CENTRAL FLORIDA INST FOR SIMULATION & TRAINING 3280 PROGRESS DR ORLANDO FL 32826
1	HIGH PERFORMANCE TECH INC VIRGINIA TO 9391 BEARDS HILL RD STE 101 ABERDEEN MD 21001

NO. OF COPIES	ORGANIZATION
	ABERDEEN PROVING GROUND
11	DIR USARL AMSRL IS (ALC) DR J GANTT R SLIFE P EMMERMAN AMSRL IS E (ALC) COL R PRICE AMSRL IS EE (ALC) R CIONCO R MEYERS AMSRL IS ES MAJ VAGLIA M A THOMAS V LONG J FORESTER E HEILMAN

INTENTIONALLY LEFT BLANK.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 1997	Summary, Jun 94 - Aug 95

4. TITLE AND SUBTITLE
An Implementation of a Dynamic Synthetic Environment Using the Silicon Graphics Performer Graphics Library

5. FUNDING NUMBERS
622618H8011

6. AUTHOR(S)
Mark A. Thomas

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRL-IS-ES
Aberdeen Proving Ground, MD 21005-5067

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-1454

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

10.SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Synthetic, virtual environments are graphical representations of physical reality that humans can interact with using a variety of computer interfaces. A synthetic, virtual environment might be a building interior, an entire city, a large land area, or even a complete fictional world. High-speed computer graphics, combined with supercomputer processing power, allow the development of high-detail, realistic environments for training and entertainment. The processing power of supercomputers allows the computation of real-world, physical phenomena such as terrain deformation, atmospheric clouds, smoke plumes, and the physical effects caused by the turbulence and irregularities in the atmosphere. Traditional graphics implementations of these phenomena require the use of low-level graphical objects, voxels, or rotating texture maps, to render these phenomena. This report presents an implementation of Fractal Ellipsoids to represent smoke plumes using the Silicon Graphics (SGI) Performer Graphics Library. In addition, the representation of deformable, changeable terrain using the ARL Variable Resolution Terrain model is explained. The applicability of these techniques provides a powerful mechanism to merge high-speed, realistic graphics rendering with the real-time data processing and computational power of supercomputers, which will provide synthetic environments with real-world characteristics.

14. SUBJECT TERMS
synthetic environments, dynamic terrain, distributed computing, real-time graphics

15. NUMBER OF PAGES
19

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500
Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

INTENTIONALLY LEFT BLANK.

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. ARL Report Number/Author ARL-TR-1454 (Thomas) Date of Report August 1997

2. Date Report Received _____

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) _____

4. Specifically, how is the report being used? (Information source, design data, procedure, source of ideas, etc.) _____

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate. _____

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.) _____

CURRENT ADDRESS

_____
Organization

_____
Name E-mail Name

_____
Street or P.O. Box No.

_____
City, State, Zip Code

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

OLD ADDRESS

_____
Organization

_____
Name

_____
Street or P.O. Box No.

_____
City, State, Zip Code

(Remove this sheet, fold as indicated, tape closed, and mail.)
(DO NOT STAPLE)